Scikit-learn Supervised Methods

> by: Saeed Mohagheghi + 🤖 AI

🧠 Classification Methods in scikit-learn

Classifier	Import Statement	Pros	Cons
Logistic Regression	`from sklearn.linear_model import LogisticRegression`	Simple, interpretable, works well for linearly separable data	Struggles with non-linear relationships
K-Nearest Neighbors (KNN)	`from sklearn.neighbors import KNeighborsClassifier`	No training phase, intuitive, good for small datasets	Slow prediction, sensitive to irrelevant features
Support Vector Machine (SVM)	`from sklearn.svm import SVC`	Effective in high-dimensional spaces, robust to overfitting	Computationally expensive, hard to tune
Decision Tree	`from sklearn.tree import DecisionTreeClassifier`	Easy to interpret, handles non-linear data	Prone to overfitting
Random Forest	`from sklearn.ensemble import RandomForestClassifier`	Reduces overfitting, handles missing data well	Slower, less interpretable than single trees
Gradient Boosting	`from sklearn.ensemble import GradientBoostingClassifier`	High accuracy, handles complex data	Long training time, sensitive to hyperparameters
AdaBoost	`from sklearn.ensemble import AdaBoostClassifier`	Boosts weak learners, good for binary classification	Sensitive to noisy data and outliers
Naive Bayes (Gaussian)	`from sklearn.naive_bayes import GaussianNB`	Fast, works well with high-dimensional data	Assumes feature independence

Classifier	Import Statement	Pros	Cons
Linear Discriminant Analysis (LDA)	`from sklearn.discriminant_analysis import LinearDiscriminantAnalysis`	Good for dimensionality reduction, interpretable	Assumes normal distribution and equal covariance
Quadratic Discriminant Analysis (QDA)	`from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis`	Handles non-linear boundaries	Requires more data, sensitive to outliers
Stochastic Gradient Descent (SGD)	`from sklearn.linear_model import SGDClassifier`	Scales well to large datasets, online learning	Requires careful tuning, sensitive to feature scaling

📊 Regression Methods in scikit-learn

Regressor	Import Statement	Pros	Cons
Linear Regression	`from sklearn.linear_model import LinearRegression`	Simple, interpretable, fast	Assumes linearity, sensitive to outliers
Ridge Regression	`from sklearn.linear_model import Ridge`	Reduces overfitting, handles multicollinearity	Requires tuning of regularization parameter
Lasso Regression	`from sklearn.linear_model import Lasso`	Feature selection, sparse models	Can underfit if regularization is too strong
ElasticNet	`from sklearn.linear_model import ElasticNet`	Combines Ridge and Lasso, flexible	Requires tuning of two parameters
Support Vector Regression (SVR)	`from sklearn.svm import SVR`	Handles non-linear regression, robust	Computationally expensive, sensitive to parameters
Decision Tree Regressor	`from sklearn.tree import DecisionTreeRegressor`	Captures non-linear patterns, easy to interpret	Prone to overfitting
Random Forest Regressor	`from sklearn.ensemble import RandomForestRegressor`	Reduces overfitting, handles complex data	Slower, less interpretable
Gradient Boosting Regressor	`from sklearn.ensemble import GradientBoostingRegressor`	High accuracy, handles non-linear relationships	Long training time, sensitive to hyperparameters

Regressor	Import Statement	Pros	Cons
AdaBoost Regressor	`from sklearn.ensemble import AdaBoostRegressor`	Boosts weak learners, good for noisy data	Can overfit, sensitive to outliers
K-Nearest Neighbors (KNN)	`from sklearn.neighbors import KNeighborsRegressor`	No training phase, good for local patterns	Slow prediction, sensitive to irrelevant features
Bayesian Ridge Regression	`from sklearn.linear_model import BayesianRidge`	Probabilistic predictions, handles multicollinearity	Assumes Gaussian priors
Theil-Sen Regressor	`from sklearn.linear_model import TheilSenRegressor`	Robust to outliers, non-parametric	Slower for large datasets
Huber Regressor	`from sklearn.linear_model import HuberRegressor`	Robust to outliers, combines linear and robust regression	Requires tuning of epsilon parameter
Stochastic Gradient Descent (SGD)	`from sklearn.linear_model import SGDRegressor`	Scales well to large datasets, online learning	Requires careful tuning, sensitive to feature scaling

---

🧠 **Note**:

- `X` is the feature matrix and `y` is the target vector.
- Most classifiers support `.predict(X_test)` for prediction.
- You can use `GridSearchCV` or `RandomizedSearchCV` for hyperparameter tuning.